

AR

Requested Patent: JP11327921A

Title:

THREAD PRIORITY CONTROLLER AND THREAD PRIORITY CONTROL METHOD
IN COMPUTER SYSTEM ;

Abstracted Patent: JP11327921 ;

Publication Date: 1999-11-30 ;

Inventor(s): WATAKABE TAKESHI;; HASEGAWA YOSHIRO ;

Applicant(s): TOSHIBA CORP ;

Application Number: JP19980130074 19980513 ;

Priority Number(s): ;

IPC Classification: G06F9/46 ;

Equivalents: ;

ABSTRACT:

PROBLEM TO BE SOLVED: To shorten the response time by preferentially delivering an execution right to a thread waiting for an event even when the thread of the same priority is during execution at the point of time of event generation when the a fixed priority specified thread waits for the event.

SOLUTION: When the thread B 102 waits for a certain event, a judgement processing is performed by a thread information judgement part 106. In the case that the thread B 102 is a thread which requests an event waiting request first, the priority is raised for one stage by a priority addition part 107. Thus, even in the case that the thread A 101 of the originally same priority as the thread B 102 which is an event waiting thread is during the execution at the point of time of the event generation, since the thread A 101 is preempted and the thread B 102 obtains the execution right, the response time is shortened.

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-327921

(43)公開日 平成11年(1999)11月30日

(51)Int.Cl.⁶
G 0 6 F 9/46

識別記号
3 4 0

F I
G 0 6 F 9/46
3 4 0 B

審査請求 未請求 請求項の数6 OL (全7頁)

(21)出願番号 特願平10-130074
(22)出願日 平成10年(1998)5月13日

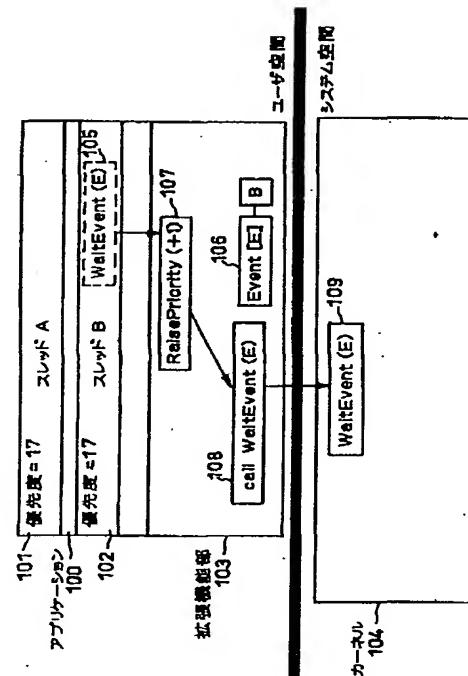
(71)出願人 000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地
(72)発明者 渡壁 健
東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内
(72)発明者 長谷川 義朗
東京都府中市東芝町1番地 株式会社東芝
府中工場内
(74)代理人 弁理士 大胡 奥夫 (外1名)

(54)【発明の名称】 計算機システムに於けるスレッド優先度制御装置、及びスレッド優先度制御方法

(57)【要約】

【課題】 固定優先度指定されたスレッドが事象待ちを行なう際、事象発生時点で同じ優先度のスレッドが実行中でも、事象待ちを行っていたスレッドに優先的に実行権を渡し、応答時間を短縮させること。

【解決手段】 スレッドB (102) がある事象待ちの際、スレッド情報判定部106により判定処理する。スレッドB (102) が事象待ち要求を最初に要求したスレッドである場合、優先度数加算部107により優先度を1段階上げる。これにより、事象発生時点でこの事象待ちスレッドであるスレッドB (102) と本来同じ優先度のスレッドA (101) が実行中の場合でも、スレッドA (101) をブリエンプションし、スレッドB (102) が実行権を得られるため、応答時間を短縮することができる。



【特許請求の範囲】

【請求項1】 スレッドに対して優先度固定の指定が可能な計算機システムのオペレーティングシステムのスレッド優先度制御装置であつて、任意の事象待ちをするスレッドにより呼び出され待っている事象が発生したらこの事象待ちをするスレッドを実行する事象待ちインターフェース手段と、上記任意の事象待ちをするスレッドの事象待ちインターフェース手段の呼び出しを横取りする事象待ちインターフェース取得手段と、上記任意の事象待ちをするスレッドが当該任意の事象の待ち要求を発行した最初のスレッドかどうかを判定するスレッド情報判定手段と、上記任意の事象待ちをするスレッドの優先度を所定の段階だけ上げる優先度変更手段と、通常呼び出される事象待ちインターフェース手段を呼び出す事象待ちインターフェース呼び出し手段とを設け、上記事象待ちインターフェース取得手段により横取りした上記任意の事象待ちをするスレッドが当該任意の事象の待ち要求を発行した最初のスレッドである場合、この横取りした上記任意の事象待ちをするスレッドの優先度を所定の段階だけ上げた後に通常呼び出されるべき事象待ちインターフェース手段を呼び出すことを特徴とする計算機システムに於けるスレッド優先度制御装置。

【請求項2】 上記優先度変更手段には、上記任意の事象待ちをするスレッドの優先度を所定の段階だけ下げる機能が設けられ、上記横取りした上記任意の事象待ちをするスレッドの優先度を所定の段階だけ上げた後に通常呼び出されるべき事象待ちインターフェース手段を呼び出した場合、事象が発生し事象待ちが終わった後に上記呼び出した事象待ちインターフェース手段から制御が戻ってきた時点で、上記横取りした上記任意の事象待ちをするスレッドの優先度を所定の段階だけ下げて元の優先度に戻すことを特徴とする請求項1記載の計算機システムに於けるスレッド優先度制御装置。

【請求項3】 上記優先度を変更する段階は1段階であることを特徴とする請求項1又は2記載の計算機システムに於けるスレッド優先度制御装置。

【請求項4】 スレッドに対して優先度固定の指定が可能な可能な計算機システムのオペレーティングシステムのスレッド優先度制御装置に於いて、任意の事象待ちをするスレッドにより呼び出され待っている事象が発生した場合、この事象待ちをするスレッドを実行する事象待ちインターフェース手段を設け、上記任意の事象待ちをするスレッドの事象待ちインターフェース手段の呼び出しを横取りし、この横取りした任意の事象待ちをするスレッドが当該任意の事象の待ち要求を発行した最初のスレッドか否かを判定し、

上記横取りした任意の事象待ちをするスレッドが当該任意の事象の待ち要求を発行した最初のスレッドである場合、上記横取りした上記任意の事象待ちをするスレッドの優先度を所定の段階だけ上げた後に本来呼び出されるべき事象待ちインターフェース手段を呼び出すようにしたこととを特徴とする計算機システムに於けるスレッド優先度制御方法。

【請求項5】 上記横取りした上記任意の事象待ちをするスレッドの優先度を所定の段階だけ上げた後に通常呼び出されるべき事象待ちインターフェース手段を呼び出した場合、事象が発生し事象待ちが終わった後に呼び出した上記事象待ちインターフェース手段から制御が戻ってきた時点で、上記横取りした上記任意の事象待ちをするスレッドの優先度を所定の段階だけ下げて元の優先度に戻すことを特徴とする請求項4記載の計算機システムに於けるスレッド優先度制御方法。

【請求項6】 上記優先度を変更する段階は1段階であることを特徴とする請求項4又は5記載の計算機システムに於けるスレッド優先度制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、優先度固定の指定が可能なコンピュータ等の情報処理装置のオペレーティングシステムに於ける固定優先度スレッドが事象待ちを行う際に優先度制御を行う優先度制御装置、即ち計算機システムに於けるスレッド優先度制御装置、及びスレッド優先度制御方法、及び計算機システムに於ける優先度制御方法に関する。

【0002】

【従来の技術】従来の情報処理装置即ち計算機システムのオペレーティングシステム(OS)の優先度制御は、固定優先度指定のスレッドが事象待ちを行う際には、事象発生時に優先度の操作を行ないものであった。その為、事象が発生した時点では、同じ優先度の別スレッドが実行中の場合、そのスレッドがCPUを放棄するまで事象待ちをしていたスレッドは実行できず、事象発生からスレッドが起動するまでの応答時間が遅延することがあった。

【0003】

【発明が解決しようとする課題】上述事項を考慮すると、ある優先順位のスレッドが実行中ということは、同じ優先順位のスレッドも実行可能ということである。従って、事象が発生した時点では、事象待ちのスレッドに実行権を渡すことが可能であれば、応答時間が向上するものである。

【0004】そこで、本発明は上記事情を考慮して成されたもので、上記不具合を解消し、固定優先度指定されたスレッドが事象待ちを行う際、事象が発生した時点と同じ優先度のスレッドが実行中でもプリエンプションして事象待ちを行っていたスレッドに優先的に実行権を渡

し、応答時間を短縮させること可能とした計算機システムに於けるスレッド優先度制御装置、及びスレッド優先度制御方法を目的とする。

【0005】

【課題を解決するための手段】本発明は上記目的を達成する為、本発明のスレッド優先度制御装置は、スレッドに対して優先度固定の指定が可能な情報処理装置のオペレーティングシステム(OS)のスレッド優先度制御装置に於いて、任意の事象待ちをするスレッドにより呼び出され待っている事象が発生したらこの事象待ちをするスレッドを実行する事象待ちインターフェース手段と、上記任意の事象待ちをするスレッドの事象待ちインターフェースの呼び出しを横取りする事象待ちインターフェース取得手段と、上記任意の事象待ちをするスレッドが当該任意の事象の待ち要求を発行した最初のスレッドかどうかを判定するスレッド情報判定手段と、上記任意の事象待ちをするスレッドの優先度を所定の段階だけ上げる優先度変更手段と、通常呼び出されるはずの事象待ちインターフェースを呼び出す事象待ちインターフェース呼び出し手段とを設け、上記事象待ちインターフェース取得手段により横取りした上記任意の事象待ちをするスレッドが当該任意の事象の待ち要求を発行した最初のスレッドである場合、この横取りした上記任意の事象待ちをするスレッドの優先度を所定の段階だけ上げた後に本来呼び出されるべき事象待ちインターフェース手段を呼び出すことを特徴とする。

【0006】このような構成によれば、固定優先度スレッドの事象待ちの際、優先度を所定の段階だけあげることで、事象が発生した時点で事象待ちスレッドと本来同じ優先度のスレッドが実行中の場合でも、ブリエンジョンし実行権を得られるため、応答時間を短縮することができる。

【0007】又、この構成の優先度変更手段に、所定の段階だけ優先度を下げる機能を設ければ、スレッドが実行権を得て処理を終了した後に元の優先度に戻すことができる。

【0008】更に、上記優先度を変更する段階を1段階とすれば、優先順位の逆転は発生せず、同じ優先度のスレッドに対してだけ、その応答時間を短縮することができる。即ち、ある優先順位のスレッドが実行中ということは、同じ優先順位のスレッドも実行可能ということであることを利用して、ある事象に対して最初に事象待ち要求をしたスレッドの優先度を上げて、その応答時間を短縮する。

【0009】

【発明の実施の形態】以下、本発明の一実施形態を図面を参照して説明する。

【0010】(第1の実施形態) 図1は、第1の実施形態に係わる計算機システム即ち情報処理装置内のオペレーティングシステム(OS)とアプリケーションソフト

ウェア(アプリケーション)の優先度制御の概念を示す為の構成図である。本実施形態の説明に必要な構成を示し、他の構成は省略している。

【0011】アプリケーション100は、情報処理装置にインストールされているアプリケーション・ソフトウェアであり、処理の基本単位である1つ以上のスレッドから成る。本実施形態では、簡単の為スレッドA(101)、スレッドB(102)から成るものとする。図示しない情報処理装置のCPUが、アプリケーション100を実行し、各部の制御を行うものである。

【0012】アプリケーション100は、拡張機能部103として、事象待ちインターフェース取得部(Event(E))105と、スレッド情報判定部(Event(E))106と、優先度数加算部(Raise Priority(+1))107と、事象待ちインターフェース呼び出し部(Call WaitEvent(E))108を持つ。

【0013】事象待ちインターフェース取得部105は、事象待ちインターフェースの呼び出しを横取りする。例えば、スレッドB(102)が事象待ちを行った際、事象待ちインターフェース取得部105によって、通常或いは本来カーネル104の事象待ちインターフェースが直接呼び出されるところを拡張機能部103を呼び出す。

【0014】事象(event)とは、アプリケーション実行中に生ずるマウスのクリックやキー入力等の外部からの介入のことである。事象が発生した時点で、その事象に対応するスレッドが実行される。

【0015】スレッド情報判定部106は、事象待ちインターフェース取得部105が横取りしたある事象の待ち要求について、当該事象の待ち要求を最初に発行したスレッドかどうかを判定し、最初に発行したスレッドであればデータ構造としてその情報をスレッド情報として格納する。このスレッド情報判定部106は、格納したスレッド情報を参照することにより、ある事象についてスレッドが待ち要求を発行した場合、それが最初かどうかを判定できる。

【0016】優先度数加算部107は、スレッドを実行するための優先度を1段階上げる。本実施形態の優先度は、優先度=17の様に数字で表わし、その数字が大きいものほど優先度が高いものとする。例えば、優先度=18は、優先度=17より優先度が高い。

【0017】本実施形態の場合、優先度の制御幅を1とし、優先度数加算部107は、優先度を1段階上げる場合、対象となる優先度の数に1を加算する。例えば、優先度=17の優先度を1段階上げる場合は、17に1加算して優先度=18とする。事象待ちインターフェース呼び出し部108は、本来の事象待ちインターフェースを呼び出す。

【0018】優先度数加算部107では、スレッド情報

判定部106が判定した情報に基づき、指定の事象待ちを要求した一番目のスレッドなら、優先順位を1段階あげて(図1ではスレッドB(102)の元の優先度は17なので、優先順位を1段階あげると優先度は18になる)、事象待ちインターフェース呼び出し部108により本来の事象待ちインターフェースを呼び出す。

【0019】カーネル104は、オペレーティングシステム(OS)の中核部分であり、メモリ管理や割り込み処理、プロセス間通信等のシステムの基本的な制御を実行する。

【0020】事象待ちインターフェース(Wait Event(E))109は、事象待ちをしているスレッドにより呼び出され、待っている事象が発生したら、この事象待ちをしているスレッドを実行する。ただし、事象が発生した場合、実行中のスレッドの優先度と事象待ちをしているスレッドの優先度を比較し、事象待ちをしているスレッドの優先度が高い場合のみ、このスレッドを実行する。実行中のスレッドの優先度が高い場合は、このスレッドの実行が終わるまで待ってから、事象待ちをしているスレッドを実行する。

【0021】次に、図2のフローチャートにより、図1の情報処理装置でのスレッドB(102)が事象待ちを行際の優先度制御の処理動作の説明をする。

【0022】スレッドB(102)がある事象待ち処理を行う(ステップA1)と、事象待ちインターフェース取得部105によって、通常或いは本来は、カーネル104の事象待ちインターフェース109が直接呼び出されるところを拡張機能部103を呼び出す(ステップA2)。

【0023】スレッド情報判定部106は、格納してある情報を参照し、この事象の待ち要求を最初に発行したスレッドとしてスレッドB(102)があるかどうかを判定する(ステップA3)。

【0024】スレッドB(102)が有った場合は(ステップA3のYes)、既にこの事象を最初に要求したスレッドとしてスレッドB(102)が認識され格納されているということなので、優先度数加算部107は、スレッドB(102)の優先度の数に1を加算し、優先度を1段階上げる(ステップA4)。即ち、スレッドB(102)の優先度=17を優先度=18とする。

【0025】次に、事象待ちインターフェースを呼び出し部108により、事象待ちインターフェースを呼び出す(ステップA5)。

【0026】この場合、スレッドB(102)の優先度は18となり、スレッドA(101)の優先度は17であるから、スレッドB(102)の優先度はスレッドA(101)より高くなる。従って、スレッドB(102)は、待っている事象が発生したら、スレッドA(102)をアリエンプションして実行権を得る。

【0027】スレッドB(102)が無かった場合は

(ステップA3のNo)、この事象の待ち要求を最初に発行した他のスレッドがある場合と、この事象待ち要求を発行したスレッドが全く無い場合が考えられる。この点について、スレッド情報判定部106で判定する(ステップA6)。

【0028】スレッドB(102)がこの事象について最初に待ち要求を発行したスレッドである場合(ステップA6のYes)、即ちこの事象待ち要求を発行したスレッドが全く無い場合、スレッド情報判定部106は、この事象待ち要求を最初にしたスレッドとしてスレッドB(102)を格納して次の処理へ移行する。

【0029】スレッドB(102)がこの事象について最初に待ち要求をしたスレッドであるから、優先度数加算部107は、スレッドB(102)の優先度の数に1を加算し、優先度を1段階上げる(ステップA4)。即ち、スレッドB(102)の優先度=17を優先度=18とする。

【0030】次に、事象待ちインターフェースを呼び出し部108により、事象待ちインターフェースを呼び出す(ステップA5)。

【0031】この場合、スレッドB(102)の優先度は18となり、スレッドA(101)の優先度は17であるから、スレッドB(102)の優先度はスレッドA(101)より高くなる。従って、スレッドB(102)は、待っている事象が発生したら、スレッドA(102)をアリエンプションして実行権を得る。

【0032】スレッドB(102)がこの事象について最初に待ち要求を発行したスレッドで無い場合(ステップA6のNo)、即ちこの事象の待ち要求を最初に発行した他のスレッドがある場合は、事象待ちインターフェースを呼び出し部108により、事象待ちインターフェースを呼び出す(ステップA5)。

【0033】この場合、スレッドB(102)の優先度は変わらず17であり、スレッドA(101)の優先度は17であるから、スレッドB(102)の優先度はスレッドA(101)と等しい。従って、スレッドB(102)は、待っている事象が発生しても、スレッドA(102)が実行していら、その処理が終わるまで実行権を得られない。

【0034】上述のように本実施形態によれば、固定優先度指定されたスレッドが事象待ちを行際、事象が発生した時点で同じ優先度のスレッドが実行中でも、このスレッドの優先度を上げ、アリエンプションして、事象待ちを行っていたスレッドに優先的に実行権を渡し、応答時間を短縮させるができる。

【0035】(第2の実施形態) 続いて、本発明に係わる第2の実施形態を説明する。図3は、情報処理装置内のオペレーティングシステム(OS)とアプリケーションソフトウェア(アプリケーション)の優先度制御の概念を示すための構成図である。本実施形態の説明に必要

な構成を示し、他の構成は省略している。

【0036】本実施形態は第1の実施形態と略構成が同じであり、優先度を1段階下げる優先度数減算部（Lower Priority（-1））208が加えられている。

【0037】アプリケーション200は、情報処理装置にインストールされているアプリケーション・ソフトウェアである。アプリケーション200は、処理の基本単位であるスレッドA（201）、スレッドB（202）からなる。図示しない情報処理装置のCPUが、アプリケーション200を実行し、各部の制御を行う。

【0038】アプリケーション200は、拡張機能部203として、事象待ちインターフェース取得部（Wait Event（E））205と、スレッド情報判定部（Event（E））206と、優先度を1段階上げる優先度数加算部（RaisePriority（+1））207と、優先度を1段階下げる優先度数減算部（Lower Priority（-1））208と、本来事象待ちインターフェースを呼び出す事象待ちインターフェース呼び出し部（Call WaitEvent（E））209を持つ。

【0039】事象待ちインターフェース取得部205、スレッド情報判定部206、優先度数加算部207、事象待ちインターフェース呼び出し部209は、前述第1の実施形態と略同様の機能なので説明を省略する。又、カーネル204と事象待ちインターフェース210も、第1の前述実施形態と略同様の機能なので説明を省略する。

【0040】本実施形態の優先度は、第1の実施形態と同様に、優先度=17の様に数字で表わし、その数字が大きいものほど優先度が高いものとする。又、優先度の制御幅も同様に1とする。優先度数加算部207は、優先度を1段階上げる為に、対象となる優先度の数に1を加算する。優先度数減算部208は、優先度を1段階下げる為に、対象となる優先度の数から1を引く。

【0041】尚、ここでは、優先度加算部と優先度減算部と二つに分けたが、優先度変更部として両方の機能を持つようにしても良いことは言うまでもない。

【0042】上記構成につき、図4のフローチャートを参照して、図3の情報処理装置でのスレッドB（202）の優先度を1段階下げる制御の処理動作の説明をする。

【0043】まず、スレッドB（202）がある事象待ち処理を行い、事象待ちインターフェース取得部205は、スレッドB（202）から事象待ちインターフェースの呼び出しを横取りする。

【0044】この事象待ちを要求した最初のスレッドならばスレッド情報判定部206に記録し、優先度数加算部207により優先度を1段階あげた後、事象待ちインターフェース呼び出し部209により通常（或いは本

来）呼び出されるべき事象待ちインターフェースを呼び出す。上記の処理動作は、前述第1の実施形態の図2での説明と同様なので詳細な説明は省略する。

【0045】この事象待ちをしている時に、事象が発生し事象待ちが終了したとする。事象待ちが終了すると、拡張機能部203内の呼び出し元である事象待ちインターフェース呼び出し部209に制御が戻る（ステップB1）。すると、最初にこの事象待ち要求を発行したスレッドかどうかをスレッド情報判定部206により判定する（ステップB2）。

【0046】スレッドB（202）が、最初にこの事象待ち要求を発行したスレッドである場合（ステップB2のYes）、優先度数減算部208は、スレッドB（202）の優先度から1を引いて優先度を下げる。この場合、スレッドB（202）の優先度は元の17から事象待ち処理時に18に上がっているので、その18から1を引き、本来の優先度=17となる。

【0047】次に、スレッドB（202）の呼び出し元である事象待ちインターフェース取得部205に制御を戻す（ステップB4）。

【0048】スレッドB（202）が、最初にこの事象待ち要求を発行したスレッドで無い場合（ステップB2のNo）、優先度は変更せず、スレッドB（202）の呼び出し元である事象待ちインターフェース取得部205に制御を戻す（ステップB4）。例えば、事象が発生した時点でスレッドA（201）が実行中だとすると、本来スレッドA（201）とスレッドB（202）は同じ優先度の為、スレッドA（201）がCPUを放棄するまでスレッドB（202）は実行権を得られない。

【0049】しかし、スレッドB（202）がこの事象待ちを要求した最初のスレッドである場合は、本実施形態によりスレッドB（202）の優先度は、優先度数加算部207により優先度を1段階上げてある為、優先度がスレッドA（201）より高くなる。従って、事象発生と共にスレッドA（201）をプリエンプションして実行権を得る。その後、事象待ちが終了すると、拡張機能部203内の呼び出し元である事象待ちインターフェース呼び出し部209に制御が戻り、図4の処理動作により優先度を下げて、スレッドB（202）とスレッドA（201）の優先度は17で同じになるが、スレッドB（202）はそのまま実行を続ける。結果、スレッドB（202）の応答時間が早くなる。

【0050】又、この場合、優先度制御幅を1段階に設定している為、優先順位の逆転は発生しない。

【0051】仮にスレッドB（202）の優先度は17で変わらず、スレッドA（201）の優先度が18であったとしたら、スレッドA（201）が実行中に事象が発生してもプリエンプションは発生しない。

【0052】スレッドA（201）が図示しない情報処理装置のCPUを放棄した後、スレッドB（202）が

実行を始める。しかし、スレッドA(201)がCPUを使用する場合は、再び優先度の高いスレッドA(201)が動作する。

【0053】スレッドB(202)がこの事象待ちを要求した最初のスレッドである場合は、本実施形態によりスレッドB(202)の優先度は、優先度数加算部207により優先度を1段階上げてあり、優先度=18となり、スレッドA(201)が図示しない情報処理装置のCPUを放棄した後、スレッドB(202)が実行を始める。事象待ちが終了し、図4の処理動作により拡張機能内で優先度を本来の値である優先度=17に下げた時点で、再び優先順位の高いスレッドA(201)が動作する。

【0054】上述のように本実施形態では、事象待ちをしているスレッドの優先度を上げて実行権を得て処理を終了した後に、元の優先度に戻すことができる。

【0055】

【発明の効果】以上詳記したように本発明によれば、固定優先度指定されたスレッドが事象待ちを行う場合、事象が発生した時点で同じ優先度のスレッドが実行中でも、事象待ちを行っていたスレッドに優先的に実行権を渡し、応答時間を短縮させることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態に係る情報処理装置の概念的構成を示す構成図。

【図2】同実施形態に係わり、優先度制御の処理動作を示すフローチャート。

【図3】本発明の第2の実施形態に係る情報処理装置の概念的構成を示す構成図。

【図4】同実施形態に係わり、優先度制御の処理動作を示すフローチャート。

【符号の説明】

100, 200…アプリケーションソフトウェア(アプリケーション)

101, 201…スレッドA

102, 202…スレッドB

103, 203…拡張機能部

104, 204…カーネル

105, 205…事象待ちインターフェース取得部

106, 206…スレッド情報判定部

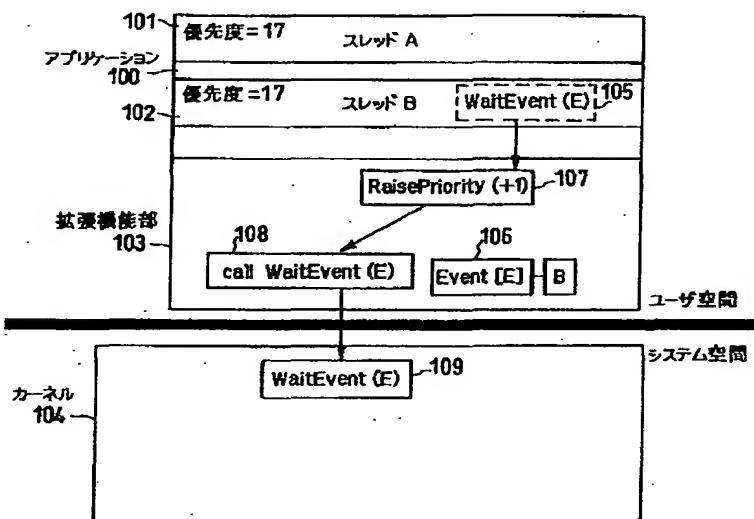
107, 207…優先度数加算部

108, 209…事象待ちインターフェース呼び出し部

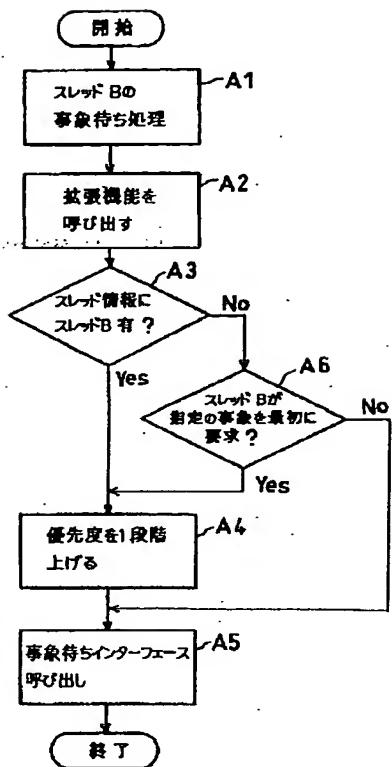
109, 210…事象待ちインターフェース

208…優先度数減算部

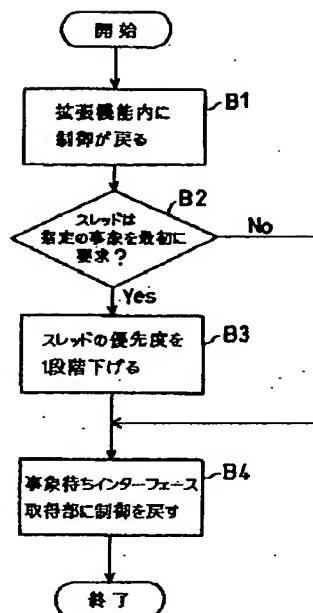
【図1】



【図2】



【図4】



【図3】

